

**PATENT**  
**5760-20600**  
**VRTS 0397**

"EXPRESS MAIL" MAILING LABEL NUMBER

EV 318248618 US

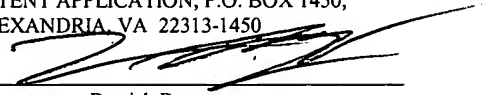
DATE OF DEPOSIT 2-6-04

I HEREBY CERTIFY THAT THIS PAPER OR  
FEE IS BEING DEPOSITED WITH THE  
UNITED STATES POSTAL SERVICE

"EXPRESS MAIL POST OFFICE TO  
ADDRESSEE" SERVICE UNDER 37 C.F.R.

§1.10 ON THE DATE INDICATED ABOVE  
AND IS ADDRESSED TO THE

COMMISSIONER FOR PATENTS, MAILSTOP  
PATENT APPLICATION, P.O. BOX 1450,  
ALEXANDRIA, VA 22313-1450

  
Derrick Brown

Restore of Backup to Computer System with Filesystem/Volume Attribute Modification

By:

Shawn L. Elahee

## **BACKGROUND OF THE INVENTION**

### **1. Field of the Invention**

This invention is related to the field of computer system backup/restore software.

### **2. Description of the Related Art**

Various software tools existing for backing up computer systems and restoring backup data to computer systems. Such tools are provided to permit recovery, at least to the most recent backup state, from user error, hardware failure in the backed-up computer system, or other disaster situations originating outside the computer system that cause data loss such as by damaging the computer system. User error may include activities such as accidental file deletion, file corruption, or erroneous file content change.

In some cases, such as certain user errors, only the affected files need be restored to the computer system. In other cases, such as hardware failure or other disaster, the entire backup state is restored to the computer system after the failure is corrected. The failure may be corrected by replacing the faulty hardware in the same computer system. In other cases, the entire computer system may be replaced. The computer system may be replaced because the faulty hardware cannot be replaced or repaired, or because it is more cost-effective to purchase a new computer system than to attempt repair of the failing computer system.

Whether hardware is replaced or the entire computer system is replaced, in some cases it is not possible to provide the same disk configuration. Over time, the readily-available fixed disk drives (or "hard" disk drives) change. For example, the capacity (or "size") of the disk drives tends to increase over time. Thus, if disk drives are being replaced in a computer system, it is likely that higher capacity disk drives will be used as compared to those being replaced. If a new computer system is being deployed, the new computer system will probably include higher capacity disk drives than the computer

system being replaced. Particularly, it is common to have fewer drives of higher capacity.

When the disk configuration (i.e., number and capacity of disk drives) is not the same between a backed-up computer system (that is, a computer system from which a backup was made) and the restore computer system (that is, the computer system to which the backup is being restored), restoration of the backup is problematic. For example, if the number of disk drives is the same between the configurations, but the new configuration has more total capacity, then space may be wasted on the new configuration due to some of the space being unallocated.

Furthermore, the storage in the original disk configuration is typically divided into various volumes onto which filesystems are mapped. Volumes may be mapped to a single disk drive or multiple disk drives. Multiple disk drives may be combined into a volume as a single logical storage. In other cases, a volume may have protection/performance attributes such as mirroring, redundant array of inexpensive disks (RAID) configurations, or striping. Such performance/protection attributes are implemented by mapping the volume to multiple disk drives. Furthermore, various filesystems may have various sizes based on the original disk configuration. If the disk configuration in the restore computer system differs from the original disk configuration, how to map volumes and filesystems to the new disk configuration may not be immediately apparent.

Some restoration tools will permit the new disk drives to be higher capacity in the restore computer system as long as the same number of disk drives are included. Such tools, during the restoration, expand the size of volumes on a given disk drive in proportion to the size difference. However, such tools are limited to use when the same number of disks are provided and are limited in flexibility. Other tools, such as the VERITAS Bare Metal Restore™ (BMR) product (version 4.5) available from VERITAS Software Corporation (Mountain View, CA) permit the user to map the volumes to a new

configuration or exclude volumes from a restore, but do not permit the volume attributes to be changed. Still other tools (such as the Intelligent Disaster Recovery Option™ (IDR) of the VERITAS Backup Exec™ product available from VERITAS Software Corporation for the Windows platform) create the volumes on the restore system, then launch

5 Microsoft's disk administration utility to permit changing various volume attributes (except for the system partition that includes the operating system software). These tools are also limited since the system partition cannot be modified. Additionally, these tools are limited in that the volume must be created on the restore computer system before it can be modified. Some volumes may not be successfully created. For example, volumes  
10 that are mapped to more than one disk drive may not be successfully created if only one disk drive is included in the restore system (e.g. striped or mirrored volumes). Finally, an administrator may manually create volumes and restore data manually. However, such a mechanism may be time consuming (both for the administrator and any users awaiting the restore) and may increase the likelihood of error.

15

### **SUMMARY OF THE INVENTION**

In some embodiments, a computer accessible medium comprises a plurality of instructions. When executed, the plurality of instructions modify at least a first attribute  
20 corresponding to a first volume, the first volume corresponding to a computer system's backup state. Subsequent to modifying the first attribute, the plurality of instructions, when executed, cause the first volume to be created on a first computer system to which the computer system's backup state is to be restored.

25 In other embodiments, a method is contemplated. At least a first attribute corresponding to a first volume is modified. The first volume corresponds to a computer system's backup state. Subsequent to the modifying, a restore of the computer system's backup state is initiated to a first computer system. The backup state includes the first volume with the modified first attribute. The restore comprises creating the first volume

on the first computer system.

In some implementations, a system comprises a first computer system and a second computer system. The first computer system is configured to modify at least a first attribute of a first volume corresponding to a computer system's backup state. Additionally, the first computer system is configured to subsequently initiate a restore of the computer system's backup state, including the first volume with the modified first attribute. The second computer system is the system on which the computer system's backup state is to be restored.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

The following detailed description makes reference to the accompanying drawings, which are now briefly described.

Fig. 1 is a block diagram of a system including several servers and a client.

Fig. 2 is a flowchart illustrating operation during one embodiment of a restore operation.

Fig. 3 is a flowchart illustrating operation of one embodiment of a mapping tool.

Fig. 4 is a block diagram of one embodiment of a computer accessible medium.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and

alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

## **DETAILED DESCRIPTION OF EMBODIMENTS**

5

Turning now to Fig. 1, a block diagram of one embodiment of a system comprising several servers and a client is shown. In the illustrated embodiment, a bare metal restore (BMR) server 10, a boot server 12, a file server 14, a backup server 16, and a client 18 are shown. The BMR server 10 is coupled to the boot server 12, the file server 14 and the backup server 16. The BMR server 10 may also be coupled to the client 18, in some embodiments. The client 18 is coupled to the boot server 12, the file server 14 and the backup server 16. Fig. 1 is illustrative of logical relationships between the servers and client. Physical connection may be established in any desired fashion (e.g. any type of network, combinations of networks, shared memory, etc.).

15

The client 18 comprises a computer system that may be restored using the servers 10, 12, 14, and 16. That is, the data to be stored on the client 18 is backed-up in a backup image 20 on the backup server 16. The data may have been backed up from the client 18 previously, and may be restored due to a failure in the client 18 (either hardware, software, or a combination thereof) or other loss of the data in the client 18. Alternatively, the data may have been backed up from another computer system (not shown) that subsequently experienced an irreparable failure or is otherwise being replaced by the client 18. The backup server 16 may implement any backup solution (e.g. the VERITAS NetBackup<sup>TM</sup> product from VERITAS Software Corporation, or any other VERITAS backup product or backup product from any other vendor).

25

At the time the backup is performed, a record is made of the system configuration. The record is illustrated as the client configuration file 26 on the BMR server 10 in Fig. 1, although the record may be stored in any form. In some embodiments, the client

configuration file 26 may be part of the backup image 20 as well, or the client configuration file 26 may be stored only as part of the backup image 20 and may be retrieved by the BMR server 10 when a client restore is to be performed. The record may be created by BMR software (e.g. either the client software executing on the client being backed up, or by the server software on the BMR server 10). The term "backup state" may be used herein to refer to the state saved from a computer system when a backup operation is performed on the computer system. The backup state may include the backup image (that is, the data backed-up from the computer system). The backup state may also include the client configuration file 26, in this embodiment. Any other state created with the backed-up data and stored for use if a restore of the backed-up data or a portion thereof is performed may be included in the backup state, in various embodiments.

The client configuration file 26 stores various information describing the client system configuration. For example, the system configuration may include the number, type, and size of storage devices in the client system. The system configuration may further include an identification of the volumes on the storage devices, including the layout of volumes on the storage devices and the attributes of the volumes. Other system configuration data may include the number and type of processors, the amount of memory, and information on other peripheral devices such as network interface hardware, printer hardware, user interface devices, etc.

The file server 14 may provide the client 18 with various software (reference numeral 22) used during the restore process, which may include the operating system software (e.g. operating system commands and libraries), BMR client software, backup client software, etc. The file server 14 may implement any file system usable over a network (e.g. network file system (NFS), Server Message Block (SMB) for Microsoft Windows<sup>TM</sup> or Session Message Block (SAMB) for Unix-like implementations, etc.). The boot server 12 may be used to provide a boot image 24 to the client 18. When the

client 18 is booted to perform a restore, the client 18 may use standard network boot protocols to boot using the boot image 24. The boot image 24 may include a customized boot procedure created by a restore tool 30 on the BMR server 10, to restore the client 18 to the state corresponding to the backup image 20 (including the data in the backup image 20, along with the configuration of the computer system at the time of the backup). In some embodiments, a media boot is supported in which the boot image 24 and software 22 are stored on a computer accessible medium such as a compact disc, and the disc is used to boot the client 18. In such embodiments, the boot server 12 and the file server 14 may be eliminated.

In one embodiment, each of the BMR server 10, the boot server 12, the file server 14, and the backup server 16 comprise computer systems configured to execute the corresponding server software. In some embodiments, one or more of the servers 10, 12, 14, and 16 may be combined onto the same physical computer system, as desired.

As part of the customized boot procedure, the configuration of the client's storage subsystem is restored. As mentioned above, in some cases, the client 18 may have a different set of storage devices than that indicated in the client configuration file 26. For example, different sets of storage devices may be encountered if the client 18 is a different physical computer system than the computer system that was backed-up to create the backup image 20, or if a storage device or devices are replaced with different devices. The restore tool 30 may permit the restoration to the different configuration. In the illustrated embodiment, the restore tool 30 includes a mapping tool 28 to allow the user to map the volumes and filesystems identified in the client configuration file 26 to the storage devices included in the client 18. Additionally, the mapping tool 28 may permit various attributes of the volumes/filesystems to be changed by the user. In some embodiments, the mapping tool 28 may execute on the BMR server 10. In other embodiments, the mapping tool 28 may execute elsewhere (e.g. on the client 18, as illustrated in dotted form in Fig. 1).



.. ..

In some embodiments, the user may be permitted to map volumes/filesystems to the client storage devices (and optionally change volume or filesystem attributes) prior to initiating the restore operation to the client 18. For example, the user may map

5 volumes/filesystems to the client storage devices prior to the creation of the volumes on the client storage devices. In some such embodiments, the volume attributes/filesystem attributes may be changed so that the volumes/filesystems may be successfully created on the client storage devices for cases in which the volumes/filesystems with the original attributes may not be created on the client storage devices.

10

In some embodiments, the mapping tool 28 accepts user modifications to the volume/filesystem mappings to client storage devices and user modifications to the volume or filesystem attributes. The mapping tool 28 may modify the client configuration file 26 to reflect the changes in attributes/mappings. In other embodiments,

15 a new client configuration file may be generated to reflect the changes. The modified client configuration file 26 or the new client configuration file may be used by the restore tool 30 to generate the customized boot procedure for the client 18. In still other embodiments, the modifications may be effected to any representation of the mapping of volumes/filesystems to storage devices and any representation of volume attributes. In

20 some embodiments, the mapping tool 28 may provide automatic mapping of volumes/filesystems to client storage devices (along with modifications of volume/filesystem attributes) in addition to or instead of accepting user input to modify the mapping/attributes.

25

While the present embodiment restores the client 18 using a customized boot procedure, any restore mechanism for creating the volumes/filesystems (with potentially modified attributes) on the client 18 and for restoring the backup image 20 into the created volumes/filesystems may be used.

In some embodiments, the mapping tool 28 may provide a command line interface to permit the user to modify volume/filesystem attributes. Other embodiments may provide a graphical user interface (GUI). Still other embodiments may provide both a command line interface and a GUI. One embodiment of the GUI is described in more detail below.

As used herein, the term "volume" may refer to any defined amount of storage on one or more storage devices. The term volume may include, for example, logical volume as used in Unix-like operating systems such as Linux, AIX from IBM Corp., Solaris from Sun Microsystems, etc. The term volume may also include, for example, a volume and/or a volume set as defined by various volume manager software such as VERITAS Volume Manager™ from VERITAS Software Corporation. The term volume may further include a partition as used, e.g., in the Microsoft Windows™ and/or disk operating system (DOS) operating systems. A partition may be a division of storage on a single disk drive, isolating the partitioned storage from other storage on the disk drive. Partitions may be primary or extended. The term volume may further refer to volume groups. The term "filesystem" may generally refer to any system for managing the organization and tracking of files that are stored on the volumes.

Generally, volumes and filesystems may have any desired attributes in various embodiments. An attribute may generally refer to any property assigned to the volume or filesystem. For example, a volume may have a size attribute indicating the size of the volume, a volume type attribute, one or more performance and/or protection attributes. A filesystem may have a size attribute, and may also have type and performance/protection attributes in some embodiments.

The volume type attribute definition may vary from operating system to operating system. For example, there are many brands of "Unix-like" operating systems (i.e. operating systems generally developed to the Unix model). In IBM's AIX, volume types

may include a journaled filesystem (JFS) and a virtual file system (VFS) type. In Hewlett-Packard's HP-UX and Sun's Solaris implementations of Unix, volume types may include boot, swap, root, etc. In Linux, volume types may include boot and swap. In the Windows operating system, volume types may include primary partitions, extended  
5 partitions, and other volumes.

Performance attributes may include any attributes that affect the performance of the volume or filesystem. For example, performance attributes may affect the latency of data access on the volume/filesystem, the bandwidth available for data access, etc. In  
10 some embodiments, the performance attributes may include whether or not the volume/filesystem is striped across two or more storage devices.

Protection attributes may include any attributes that affect the fault-tolerance of data on the volume/filesystem. For example, the protection attributes may include  
15 whether or not a volume/filesystem is mirrored (protecting against the failure of one of the storage devices to which data is mirrored). The protection attributes may include various RAID levels that may be used on a volume or filesystem.

Permitting the modification of volume/filesystem attributes may provide the user  
20 with flexibility in restoring data to the client 18. For example, the user may increase or decrease the size of a volume or filesystem to fit on the storage devices included in the client 18. The user may modify performance/protection attributes to fit on the storage devices as well. For example, if mirroring or striping of a volume is enabled and there  
25 are not enough storage devices to support the mirroring or striping, the mirroring or striping may be disabled. Similarly, the RAID level may be modified. On the other hand, the user may also modify protection/performance attributes to increase performance and/or protection, if the client storage devices permit. For example, striping or mirroring may be enabled, or the RAID level may be increased.

As mentioned above, in some embodiments, the mapping tool 28 may include a GUI for making volume/filesystem changes. In one embodiment, the GUI may include a graphical display of the storage system configuration from the backed-up system (referred to as the "original configuration") and a storage system configuration developed by the user (referred to as the "new configuration"). Both the original configuration and the new configuration may be visible to the user. For example, the original configuration and the new configuration may be displayed in separate panels within the same window. Alternatively, the original configuration and the new configuration may be displayed in separate windows, which may be moved, resized, etc. separately and which may be permitted to overlap each other on the display screen. Additionally, the GUI may also include one or more mechanisms for capturing data from the original configuration and inserting the data in the new configuration. Various capturing mechanisms may be supported. For example, the mapping tool 28 may support one or more of a "drag and drop" mechanism, a "copy and paste" mechanism, the launch of a "wizard" to guide the user through a set of steps to accomplish the capture and insert, etc.

As used herein, a storage device may include any sort of computer storage media such as, for example, fixed or removable disk drives, external disk drives (including solid state drives that comprise non-volatile memory such as Flash memory) coupled via a peripheral interface, tape drives, etc.

Turning next to Fig. 2, a flowchart is shown illustrating operation during a restore, including operation of one embodiment of the restore tool 30. In some embodiments, the restore tool 30 is implemented in software. In such embodiments, the restore tool 30 may include a plurality of instructions which, when executed, implement the operation assigned to the restore tool 30 in the flowchart of Fig. 2.

The restore tool 30 may determine if the new configuration (of the client 18) is the same as the original configuration (corresponding to the backup state that is to be

restored). For purposes of this disclosure, the restore tool 30 may determine that the new configuration and the original configuration are the same if the storage configuration is the same (e.g. same number, type, and size of storage devices). In other embodiments, additional checks may be made to determine if the new configuration and the original  
5 configuration are the same, as desired. If the new configuration is the same as the original configuration (decision block 40, "yes" leg), mapping of the volumes and filesystems may not be needed and the mapping tool may not be invoked. That is, if the new configuration and the original configuration are the same, the client configuration file 26 may be used as-is to restore the client 18. If the new configuration differs from the  
10 original configuration (decision block 40, "no" leg), the mapping tool may be invoked to map the volumes/filesystems to the client 18 (block 42). Operation of one embodiment of block 42 is shown in Fig. 3 and described in more detail below.

The restore tool 30 may generate the restore procedure for the client 18 (e.g. part  
15 of the boot image 24, in the embodiment of Fig. 1) using the client configuration file 26 (as modified by the mapping tool 28, if the new configuration and the original configuration are not the same) (block 44). The client may then be restored (block 46). Restoration of the client 46 may occur, in this embodiment, by the client 18 booting and executing the boot image 24.

20

Block 46 is shown in exploded view in Fig. 2. Among other things, restoring the client 18 may include creating the volumes on the client, and creating the filesystems on the volumes (block 48). Any modifications made by the mapping tool 28 to the volume attributes and/or filesystem attributes are reflected in the created volumes/filesystems.  
25 Subsequently, the data from the backup image 20 is restored to the volumes/filesystems created in block 48 (block 50). Other operations may intervene between blocks 48 and 50, in various embodiments, as desired.

It is noted that, while Fig. 2 illustrates invoking the mapping tool 28 in response

to detecting dissimilar configurations, a user may also be permitted to invoke the mapping tool 28 directly. For example, if the new configuration includes more storage than the original configuration, the user may wish to increase the performance/protection of various volumes and/or filesystems by modifying various attributes.

5

Turning now to Fig. 3, a flowchart is shown illustrating operation of one embodiment of the mapping tool 28 (block 42 in Fig. 2). In some embodiments, the mapping tool 28 is implemented in software. In such embodiments, the mapping tool 28 may include a plurality of instructions which, when executed, implement the operation assigned to the mapping tool 28 in the flowchart of Fig. 3.

10

The mapping tool 28 may obtain the original and new configurations (block 60). For example, in the illustrated embodiment, the original configuration may be defined in the client configuration file 26. The new configuration may be obtained from the client 18. In some embodiments, the mapping tool 28 may be supplied with a configuration file similar to the client configuration file 26 that describes the storage devices in the client 18 (e.g. type and size). In other embodiments, the mapping tool 28 may perform a discovery process on the client 18 or may cause a discovery process to be performed, and may receive the results. The discovery process may identify the hardware in the client 18, including various storage devices. In some embodiments, the discovery process may include a discovery boot.

15

20

The mapping tool 28 may permit the user to drag and drop volumes into the new configuration as desired (not shown in Fig. 3). Additionally, the mapping tool 28 may permit the user to modify various attributes of the volume (or a filesystem thereon). If the user selects a volume or filesystem (decision block 62, "yes" leg), the mapping tool 28 may display the attributes of the volume/filesystem (block 64). The mapping tool 28 may accept any user modifications to the attributes (block 66).

25

In some embodiments, the mapping tool 28 may provide at least some validation checking on the modifications (decision block 68). For example, the mapping tool 28 may verify that the mapping of volumes to storage devices in the new configuration meets any operating-system specific requirements (if any), that the mapping of volumes to storage devices does not exceed the storage capacity on the storage devices, that the mapping of a volume to two or more storage devices (if any) meets any requirements for such mappings, and/or that any changes to the volume implemented by the user as part of inserting the volume into the new configuration are permissible. An example of operating-system specific requirements, in clients using the Microsoft Windows<sup>TM</sup> operating system, may include a requirement that partitions and other types of volumes are not mixed on the same storage device and a requirement that a maximum of 4 primary partitions are included on the same storage device. An example of verifying mapping of a volume to two or more storage devices may include, for a striped volume, that each stripe is mapped to a different physical storage device. Similar constraints may exist for mirrored volumes or RAID volumes, in some embodiments. An example of verifying the changes to the volume may include, if a user reduces the size of the volume, that the reduced size is not smaller than the amount of data actually stored in the volume. Any set of validation checking may be implemented in various embodiments. For example, some validation checking may be delayed until the user commits the changes (not shown in Fig. 3). Additionally, not all validation checking may be performed in response to accepting user input. In some embodiments, no validation checking may be performed.

If the validation checking detects an invalid modification (decision block 68, "no" leg), the mapping tool 28 may inform the user of the problem (block 70) and may return to accept more user input (to correct the problem – block 66). If the validation checking detects valid modifications (decision block 68, "yes" leg), the mapping tool 28 may await the selection of another volume or filesystem (decision block 62).

The user may select as many volumes/filesystems as desired and may modify

various attributes as described above for blocks 62, 64, 66, 68, and 70. At some point, the user finishes and may indicate that the configuration is complete and ready to be committed (decision block 72). In the present embodiment, the user may enter modification via the GUI and may view the proposed modifications and their effect on the new configuration. Once the modifications are complete, the modifications may be committed to the new client configuration and the restore on the client 18 may be started. If the user commits the modifications (decision block 72, "yes" leg), the mapping tool 28 may update the client configuration file (or generate a new configuration file) and exit (e.g. to block 44 in Fig. 2) (block 74).

The embodiment of Fig. 3 provides an interactive user interface for accepting user input. For example, the embodiment of Fig. 3 may be used as part of a GUI implementation of the mapping tool 28. There may also be command line implementations of the mapping tool 28. For example, the command line implementation may include accepting user modifications (block 66), validating them (decision block 68), and either displaying an error (block 70) or implementing the requested modification in the client configuration file if no error is detected. In some embodiments, the command line interface may permit changes to be created followed by a commit of the changes, similar to the flowchart of Fig. 3.

Turning now to Fig. 4, a block diagram of a computer accessible medium 150 is shown. Generally speaking, a computer accessible medium may include any media accessible by a computer during use to provide instructions and/or data to the computer. For example, a computer accessible medium may include storage media such as magnetic or optical media, e.g., disk (fixed or removable), CD-ROM, or DVD-ROM, CD-R, CD-RW, DVD-R, DVD-RW, volatile or non-volatile memory media such as RAM (e.g. synchronous dynamic RAM (SDRAM), Rambus DRAM (RDRAM), static RAM (SRAM), etc.), ROM, Flash memory, non-volatile memory (e.g. Flash memory) accessible via a peripheral interface such as the Universal Serial Bus (USB) interface,



etc., as well as media accessible via transmission media or signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as a network and/or a wireless link. The computer accessible medium 150 in Fig. 4 may be encoded with the restore tool 30 (including the mapping tool 28). The restore tool 30  
5 and/or the mapping tool 28 may comprise instructions which, when executed, implement the functionality described herein for the software. Generally, the computer accessible medium 150 may store any set of instructions which, when executed, implement a portion or all of the flowcharts shown in one or more of Figs. 2-3. In some embodiments, a computer accessible medium similar to the computer accessible medium 150 may be  
10 included in a client 18 and/or the BMR server 10.

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

15